Chapter 16

Emergent Semantics from Media Blending

Edward Altman, Institute for Infocomme Research, Singapore

Lonce Wyse, Institute for Infocomm Research, Singapore

ABSTRACT

The computation of emergent semantics for blending media into creative compositions is based on the idea that meaning is endowed upon the media in the context of other media and through interaction with the user. The interactive composition of digital content in modern production environments remains a challenging problem since much critical semantic information resides implicitly within the media, the relationships between media models, and the aesthetic goals of the creative artist. The composition of heterogeneous media types depends upon the formulation of integrative structures for the discovery and management of semantics. This semantics emerges through the application of generic blending operators and a domain ontology of pre-existing media assets and synthesis models. In this chapter, we will show the generation of emergent semantics from blending networks in the domains of audio generation from synthesis models, automated home video editing, and information mining from multimedia presentations.

INTRODUCTION

Today, there exists a plethora of pre-existing digital media content, synthesis models, and authored productions that are available for the creation of new media productions for games, presentations, reports, illustrated manuals, and instructional

materials for distance education. Technologies from sophisticated authoring environments for nonlinear video editing, audio synthesis, and information management systems are increasingly finding their way into a new class of easy to use, partially automated, authoring tools. This trend in media production is expanding the life cycle of digital media from content-centric authoring, storage, and distribution to include user-centric semantics for performing stylized compositions, information mining, and the reuse of the content in ways not envisioned at the time of the original media creation. The automation of digital media production at a semantic level remains a challenging problem since much critical information resides implicitly within the media, the relationships between media, and the aesthetic goals of the creative artist. A key problem in modern production environments is therefore the discovery and management of media semantics that emerges from the structured blending of pre-existing media assets. This chapter introduces a model-based framework for media blending that supports the creative composition of media elements from pre-existing resources.

The vast quantity of pre-existing media from CD's, the Internet, and local recordings that are currently available has motivated recent research into automation technologies for digital media (Davis, 1995; Funkhouser et al., 2004; Kovar & Gleicher, 2003). Traditional authoring tools require extensive training before the user becomes proficient and normally consume enormous time to compose relatively simple productions even by skilled professionals. This contrasts with the needs of the non-professional media author who would prefer high level insights into how media elements can be transformed to create the target production, as well as tools to automate the composition from semantically meaningful models. Such creative insights arise from the ability to flexibly manipulate information and discover new relationships relative to a given task. However, current methods of information retrieval and content production do not adequately support exploration and discovery in mixed media (Santini, Gupta, & Jain, 2001). A key problem for media production environments is that the task semantics for content repurposing depends upon both the media types and the context of the current task. In this chapter we claim that many semantics based operations, including summarization, retrieval, composition, and synchronization can be represented as a more general operation called, *media blending*. Blending is an operation that occurs across two or more media elements to yield a new structure called, the *blend*. The blend is formed by inheriting partial semantics from the input media and generating an emergent structure containing information from the current task and the source media. Thus the semantics of the blend emerges from interactions among the media descriptions, the task to be performed, and the creative input of the user.

Automated support for managing the semantics of media content would be beneficial for diverse applications, such as video editing (Davis, 1995; Kellock & Altman, 2000), sound synthesis (Rolland & Pachet, 1995), and mining information from presentations (Dorai, Kermani, & Stewart, 2001). A common characteristic among these domains that will be emphasized in this chapter is the need to manage multiple media sources at the semantic level. For sound production, there is a rich set of semantics associated with sound effects collections and audio synthesis models that typically come with semantically labeled control parameters. In the case of automatic home video editing, the control logic is informed by the relationships between music structure and video cuts as described in film theory to yield a production with a particular composition style (Sharff,

1982). In the case of presentation mining from e-learning content, there is an association between pedagogical structures within a lecture video and other content resources, such as textbooks and slide presentations, that can be used to inform a search engine when responding to a student's query. In each case, the user-centric production of media involves the dynamic blending of information from different media. In this chapter, we will show the construction of blending networks for user-centric media processing in the domains of audio generation from sound synthesis models, automated home video editing, and presentation mining.

BACKGROUND

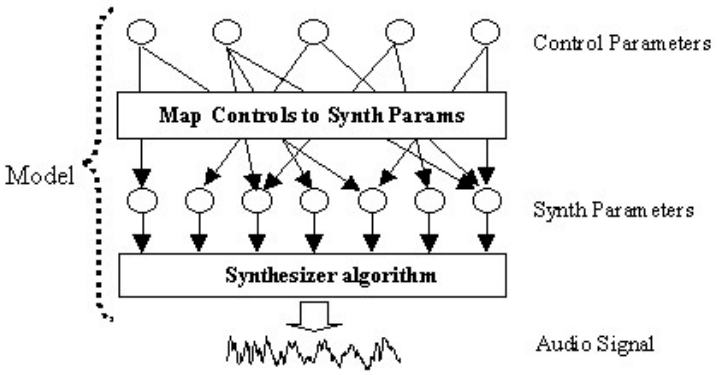
Models are fundamental for the construction of blending networks (Veale & O'Donoghue, 2000). Blending networks have their origins in frame based reasoning systems and have recently been applied in cognitive linguistics to link discourse analysis with fundamental structures of cognition. According to Conceptual Integration Theory from cognitive linguistics, thought and language depend upon our capabilities to manipulate webs of mappings between mental spaces (Fauconnier, 1997). These mental space mappings form the basis for the understanding of metaphors and other forms of discourse as conceptual blends. Similarly, the experiential qualities of media constitute a form of discourse that can only be understood through the creation of deep models of media (Staab, Maedche, Nack, Santini, & Steels, 2002). Prior work on conceptual blending provides a theoretical framework for the extension of blending theory to digital media. Consequently, audio perception, video appreciation, and information mining may be viewed as a form of media discourse. Accordingly, the claim in this chapter is that principles of conceptual blending derived from analysis of language usage may also be applied to the processing of media. In the remainder of this section we will describe three scenarios for media blending, then review the literature on conceptual blending and metaphor. The following sections will relate these structures to concrete examples of media blending.

Audio Models

Intricate relations between audio perception and cognition associated with sound production techniques pose interesting challenges regarding the semantics that emerge from combinations of constituent elements. The semantics of sound tend to be more flexible than the semantics associated with graphics and have a more tenuous relationship to the world of objects and events than do graphics. For example, the sound of a crunching watermelon can be indicative of a cool refreshing indulgence on a summer day, or it can add juicy impact to a punch for which the sound is infamously used in film production. The art of sound effects production depends heavily on the combination, reuse, and recontextualization of libraries of prerecorded material. Labeling sounds in a database in a way that supports reuse in flexible semantic contexts is a challenge.

A current trend in audio is the move toward structured representations (Rolland & Pachet, 1995) that we will call "models". A sound model is a parameterized algorithm for generating a class of sounds as shown schematically in Figure 1. Models are useful in media production partly because of their low memory/bandwidth requirements, since it

Figure 1. A "sound model" includes a synthesis algorithm capable of generating a specific range of sounds, and parameters that determine the behaviour of the model



takes much less memory to parameterize a synthesis model than it does to code the raw audio data. Also, models meet the requirement from interactive media, such as games, that audio be generated in real time in response to unpredictable events in an interactive environment.

The sound model design process involves building an algorithm from component signal generators and transformers (modulators, filters, etc.) that are patched together in a signal flow network that generates audio at the output. Models are designed to meet specifications on a) the class of sounds the model needs to cover and b) the method of controlling the model through parameters that are exposed to the user. Models are associated with semantics in a way that general-purpose synthesizers are not, because they are specialized to create a much narrower range of sounds. They also take on semantics by virtue of the real-time interactivity — they have "responsive behaviors" in a way that recorded sounds do not.

As media objects, models present interesting opportunities and challenges for effective exploitation in graphical, audio, or mixed media. A database of sound models is different from a database of recorded sounds in that the accessible sounds in the database are (i) not actually present, but potential and (ii) infinite in variability due to the dynamic parameterization that recorded sounds do not afford. Model building is a labor intensive job for experts, so exploiting a database of pre-existing sound models potentially has tremendous value.

Another trend in audio, as well as other forms of digital media, is to attempt to automatically extract semantics from raw media data. The utility of being able to identify "a baby crying" or a "window breaking" in an audio stream should be self-apparent, as should the difficulty of the task. Typically, audio analysis is based on adaptive association between low-level signal features (such as spectral centroid, basis vectors, zero crossings, pitch, and noise measures) and labels provided by a "supervisor", or based on an association with data from another media stream such as video. The difficulty

lies in the fact that there is no such thing as "the" semantics, and any semantics there may be, are dependent upon contexts both in and outside of the media itself. The human-in-the-loop and the intermediate representations between physical attributes and deep semantics that models offer can be effective bridges across this gap.

Video Editing

The blending of two or more media that combines perceptual aspects from each media to create a new effect is a common technique in film production. In film editing, the visual presentation of the scene tells the story from the character's point of view. Music is added to convey information about the character's emotional state, such as fear, excitement, calm, and joy. Thus, when the selection of video cut points along key visual events are synchronized with associated features in the music, the audience experiences the blended media according to the emergent semantics of the cinematic edit (Sharff, 1982).

The non-professional media author of a home video may know what style of editing they prefer, but lack the detailed knowledge, or time, to perform the editing operations. Similarly, they may know what music selections to add to the edited video, but lack the tools and insight to match the beat, tempo, and other features from the music with suitable events in the video content. The challenge for semi-automated video editing tools is to combine the stylistic editing logic with metadata descriptions of the selected music and video, then opportunistically blend the source media to create the final product (Kellock & Altman, 2000; Davis, 1995).

Presentation Mining

The utilization of media semantics is important not only for audio synthesis and video editing, but also for information intensive tasks, such as composing and subsequently mining multimedia presentations. There are a rapidly growing number of corporate media archives, multimedia presentations, and modularized distance learning courseware which contain valuable information that remains inaccessible outside the original production context. For instance, a common technique for authoring modular courseware is to produce a series of short, self contained multimedia presentations for topics in the syllabus, then customize the composition of these elements for the target audience (Thompson Learning, n.d.; WebCT, n.d.). The control logic for the sequencing and navigation through the course content is specified through the use of description languages. However this normally does not include a semantic description of pedagogical events, domain models, or dependencies among media resources that would aid in the exploration of the media by the user. Once the course is constructed, it becomes very difficult to modify or adapt the content to new contexts.

Recorded corporate presentations and distance learning lectures are notoriously difficult to search for information or reuse in a different context. This difficulty arises from the fact that the semantics of the presentation is fixed at the time of production. The media blending framework is designed to support the discovery and generation of emergent semantics through the use of ontologies for modeling domain information, composition logic, and media descriptions.

MEDIA BLENDING FRAMEWORK

The key issue of this chapter is to empower the media producer to more easily create complex media assets by leveraging control over emergent semantics derived from media blends. Current media production techniques involve the human interaction with collections of media libraries and the use of specialized processing tools, but they do not yet provide support for utilizing semantic information in creative compositions. The development of standards, such as mpeg-7, AAF, and SMIL, facilitate the description, shared editing, and structured presentation of media elements (Nack & Hardman, 2002). The combination of description languages and rendering engines for sound (C-Sound, MAX), and video (DirectShow, QuickTime) provide powerful tools for composing and rendering media after it is produced. Recent efforts toward automated media production (Davis, 1995; Kellock & Altman, 2000) begin to demonstrate the power of model-based tools for authoring creative compositions. These approaches depend upon proprietary methods and tend to work only in specialized contexts. The objective of media blending is to unify these disparate approaches under a common framework that results in more efficient methods for managing media semantics.

In this section we will motivate the need for a media blending framework by citing current limitations in audio design methods, then provide an illustrative example of an audio blending network. This section concludes with a concise formulation of media blending.

Sound Semantics in Audio Production

Production houses typically have hundreds of CDs of sound effect material stored in databases and access the audio by searching an index of "semantic" labels. A fragment of a sound effects database shown in Table 1 illustrates that sounds typically acquire their semantics from the context in which they were recorded.

One thing that makes it difficult to repurpose sounds from a database labeled this way is that sounds within a category can sound very different, and sounds in different categories can sound very similar. That is, the sounds in production libraries are generally not classified by clusters of acoustic features. Instead, there are several classes of semantic descriptors typically used for sound:

- **Sources as descriptors.** *Dog* barking, *tires* screeching, *gun* shot. The benefit of using sources as semantic descriptors is that the descriptions come from lay language that everybody speaks. Sources are very succinct descriptions and come with a rich set of relationships to other objects that we know about. A drawback to sources as descriptors is that some sounds have no possible, or at least obvious, physical cause (e.g., the sound of an engine changing in size). Even if a physical source is responsible for a sound, it may be impossible to identify. Similarly, any given sound may have many unrelated possible sources. Finally, a given source can have acoustically unrelated sounds associated with it, for example, a train generates whistles, steam, rolling, and horn sounds.
- **Actions and events as descriptors.** Dog *barking*, tires *screeching*, gun *shot*. Russolo's early musical noise machines, or *intonurumori*, had onomatopoetic names allied with actions, including *howler, roarer*, *crackler*, *rubber*, *hummer*, *gurgler*, *hisser*, *whistler*, b*urster*, c*roaker*, and r*ustler* (Russolo, 1916). The benefit

Table 1. Semantic descriptions in a database of common sound effects

Title	Description	Duration
Train, steam	1900 steam train interior: start, run, stop	3:02
Train, steam	Steam train whistle: long blast, close up	:12
Train steam	Steam train whistle: several blasts, close up	:10
Train, steam	English steam locomotive whistle: short blast	:03
Train, steam	English steam locomotive whistle: short toot	:02
Train, subway	Platform ambience: train arrives, departs	1:28
Train, subway	Platform ambience: train arrives, departs, P.A.	1:24
Train, subway	External: pull into station, pass by, stop, exit station	1:42
Train, subway	External: pull into station, stop, exit station	1:08
Train, subway	External: pull away: some squeak, wheel clacking	1:05
Train, subway	Interior of train: traveling between stops	3:00

of actions and events as descriptors is that they can often be assigned even when source identification is impossible (a screech is descriptive whether the sound is from tires or a child). Actions and events are also familiar to a layperson for describing sounds (scraping, falling, pounding, screaming, sliding, rolling, coughing, clicking). A drawback is that in some cases it may be difficult or impossible for sounds to be described this way. Unrelated sounds can also have the same description in terms of actions and events. Finally, the description can be quite subjective — one person's "gust" is another's "blow".

- **Source attributes as sound descriptors.** *Big* dog, *metal* floor, *hollow* wood. Such descriptions are often easier to obtain than source identification and are still useful even when source identification is impossible. These attributes are often scalar, which makes them quantitative and easier to deal with for a computer. The drawbacks are that it may be difficult to assign attributes for some sounds, many sounds may have the same attributes, and the assignment can be quite subjective.

Sounds may also "belong together" simply because they frequently co-occur in the environment or in man-made media. A "beach sounds" class could include crashing waves, shouting people, and dogs barking. Loose categories such as "indoor" and "outdoor" are often useful — especially in media production. A recording of a dog barking indoors would be useless for an outdoor scene.

When producers have the luxury of a high budget and are creating their own sound effects, sounds are typically constructed from a combination of recorded material, synthetic material, and manipulation. Typical manipulation techniques include time reversal, digital effects, such as filtering, delay, pitch shifting, and overlaying many different tracks to create an audio composite. To achieve the desired psychological impact for an event with sound, it is often the case that recordings of actual sounds generated by the real events are useless. For example, the sounds of a real punch or a real gun firing are entirely inadequate for creating the impression of a punch or a gun shot in cinema. The sounds that one might want to use as starting material to construct the effects come from unrelated material with possibly unrelated semantic labels in the stored database (Mott, 1990).

Semantic labels tend to commit a sound in a database to a certain usage unless the database users know how to work around the labels to suit their new media context. On the other hand, low-level physical signal attributes are not very helpful at providing human-usable knowledge about a sound, either. In the 1950's, Pierre Schaeffer made a valiant attempt at coming up with a set of generic source-independent sound descriptors. Rough English translations of the descriptors include *mass*, *dynamics*, *timbre*, *melodic profile*, *mass profile*, *grain*, and *pace*. He hoped that any sound could be described by a set of values for these descriptors. He was never satisfied with the results of his taxonomical attempts.

More recently, Dennis Smalley has developed his theory of "Spectromorphology" (Smalley, 1997) with terms that are more directly related to aural perception: *onsets* (departure, emergence, anacrusis, attack upbeat, downbeat), *continuants* (passage, transition, prolongation, maintenance, statement), *terminations* (arrival, disappearance, closure, release, resolution), *motions* (push/drag, flow, rise, throw/fling, drift, float, fly), and *growth* (unidirectional, such as ascent, planar, descent — and reciprocal, such as parabola, oscillation, undulation). This terminology has had some success in the analysis of electroacoustic works of music, which are notoriously difficult due to the unlimited sonic domain from which they draw their material and because of the lack of definitive reference to extra-sonic semantics. In fact, the clearest limitation of spectromorphology is its inability to address the referential dimension of much contemporary music.

The Blending Network

Sound models are endowed with semantics in their parameterization since the models are built for a specific media context — to cover a certain range of sounds and manipulate perceptually meaningful features under parametric control. Thus, models are given names such as "footsteps", and parameters are given names such as "walker's weight", "limp", and perhaps a variety of characteristics concerning the walking surface. If one knew the parameter values used for a particular footstep model to create a certain sound, then one could interpret the semantics from the parameter names and their values as, for example, "a heavy person walking quickly over a wet gravel surface". A single model could be associated with a wide range of different semantics depending upon the assumed values of the parameters.

In the context of audio production, models have one clear advantage over a library of recorded sounds in that they *permit their semantics to be manipulated*. Rarely does a sound producer turn to a library of sounds and use the sound unadulterated in a new media production. Recordings lend themselves to "standard" audio manipulation techniques — they can be layered, time-altered, put through standard "effects processors" such as compressors and limiters, phateners, harmonizers, etc. However, models, by design, give a sound designer handles on the sound semantics, at least for the semantics they were designed to capture. With only a recording, and no generative model, it would be difficult, for example, to change a walk into a run because in addition to the interval between the foot impacts there are a myriad of other differences in the sound due to heel-toe timing and the interaction between the foot and the surface characteristics. In a good footsteps model, the changes would all be a coordinated function of parameters controlling speed and style.

The 'footsteps' model is used in the following example to illustrate the central principles of media blending networks. Consider an audio designer who has been given the task of creating the sound of two people passing on a stairway. In the model library there are separate models for a person going up the stars and for going down the stars, but there is no model for two people passing. The key moment for the audio designer is the event when two people meet on the stairway so that both complete the step at the same time. This synchronization is not a part of either input model, but it has a semantic meaning that is crucial for the overall event.

The illustrations of blending networks use diagrams to represent the models and relationships. In these diagrams, models are represented by circles; parameters by points in the circles; and connections between parameters by lines. Each model may be realized as a complex software object that can be modified at the time the blending network is constructed. Thus the sound designer would use a high level description language to specify the configuration of the models and their connections. The configuration description is then compiled into the blending network which could then be 'run' to produce the desired sound.

The Footstep network contains two input models corresponding to the audio model for walking up the stairs and the model for walking down the stairs. Each model in Figure 2 is distinct, however they have semantically similar parameters. The starting time for climbing the stairs is t_1, the starting time for descending the stairs is t_2, the person going up is p_1, and the person going down is p_2.

The two audio models have parameters that are semantically labeled. The cross-model mapping that connects corresponding parameters in the input models is illustrated by dashed lines in Figure 3. In addition to the starting times, t_i, and the persons, p_i, that are specified explicitly in the input models, connections are established between other similar pairs of parameters, such as walking speed, s_i, and location, l_i.

The two input models inherit information from an abstract model for walking that includes percussion sounds, walking styles, and material surfaces. This forms a generic model that expresses the common features associated with the two inputs. The common features may be simple parameters, such as start time, person, speed, and location as in Figure 4. More generally, the generic model may be used to specify the components and relationships in more complex models as a domain ontology, as we shall see later.

The blending framework in Figure 5 contains a fourth model which is typically called "*the blend.* The two stair components in the input models are mapped onto a single set of stairs in the blend. The local times, t_1 and t_2, are mapped onto a common time t' in the blend. However, the two people and their locations are mapped according to the local time of the blend. Therefore, the first input model represents the audio produced while going up the stairs, whereas the second model represents the audio produced while going down. The projection from these input models onto the blend preserves time and location.

The Footstep network exhibits in the blend model various emergent structures that are not present in the inputs. This emergent structure is derived from several mechanisms available through the dynamic construction of the network. For example, the composition of elements from the inputs causes relations to become available in the blend that do not exist in either of the inputs. According to this particular construction, the blend contains two moving individuals instead of the single individual in each of the inputs. The individuals are moving in opposite directions, starting from opposite ends of the stairs,

Figure 2. Input models for the Footstep blending network

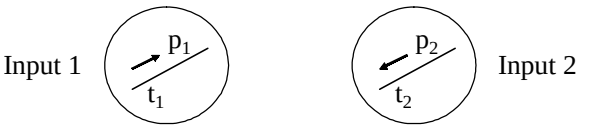


Figure 3. Cross-model mapping between the input footstep models

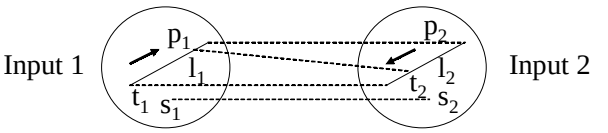
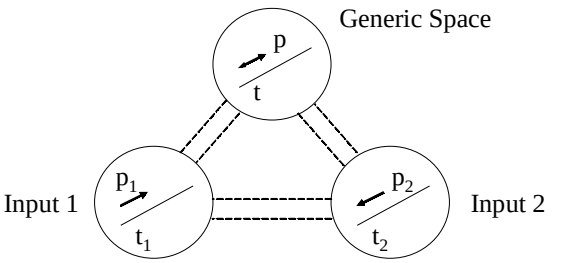


Figure 4. Inclusion of the generic model for the Footstep input models



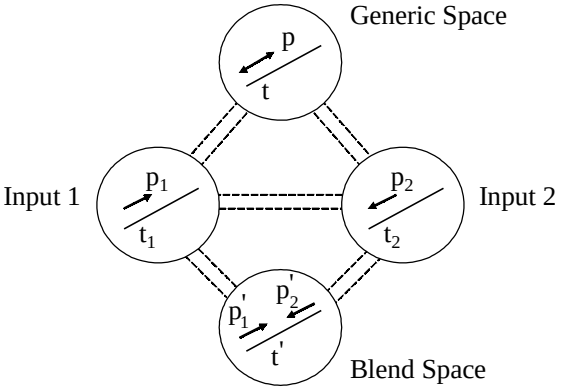
and their positions and relative temporal patterns can be compared at any time that they are on the stairs.

At this point the construction of the blending network is complete and constitutes a meta-model for the two people walking in opposite directions on the same stairs. Since this is a generative model, we can now run the scenario dynamically. In the blend there is new structure: There is no encounter in either of the input models, but the blend contains the synchronized stepping of the two individuals. The input models continue to exist in their original form, therefore information about time, location, and walking speed in the blend space can be projected back to the input models for evaluation there. This final configuration with projection of the blend model back to the input models is illustrated in Figure 5.

Blending Theory

Blending is an operation that occurs across two or more input spaces to yield a new space, the *blend*. The blend is formed by inheriting partial structure from the input spaces

Figure 5. Space mapping for the blending model in the integrated footstep network



and generating an emergent structure containing information not explicitly present in either input. A blend involves the use of information from two sources such that there are two sets of bindings that map between the input spaces and from the input spaces to the blend space. Computationally, the blending network constitutes a double scope binding between the pair of inputs and the blend space (Fauconnier, 1997). The double scope binding configuration for creating a blend is illustrated in Figure 5 and is composed of the following elements:

- **Input Spaces:** a pair of inputs, I_1 and I_2, to the network along with the models for processing the inputs. In the Footstep network, the inputs were the audio models for generating the footstep sounds.
- **Cross Space Mapping:** direct links between corresponding elements in the input spaces I_1 and I_2 or a mapping that relates the elements in one input to the corresponding elements in the other.
- **Generic Space:** defines the common structure and organization shared by the inputs and specifies the core cross-space mapping between inputs. The domain ontology for the input models is included in the generic space.
- **Blend Space:** information from the inputs I_1 and I_2 is partially projected onto a fourth space containing selective relations from the inputs. Additionally, the blend model inherits structure from the ontologies used in the generic model, as well as specific functions derived from the context of the current user task. The two footstep models were integrated into a single blend model with projection of parameter values.
- **Emergent Structure:** The blend contains new information not explicitly present in the inputs that becomes available as a result of processing the blending network. In the Footstep network, the synchronization of the footsteps at the meeting point in the blend model was the key emergent structure.

The key contribution of Conceptual Integration Theory has been the elaboration of a mechanism for double scope binding for the explanation of metaphor and the

processing of natural language discourse. The basic diagram for the double scope binding from the two input models onto the blend model was previously illustrated in Figure 5. This network is formed using the generic model to perform cross-space mapping between the two input models, then projecting selected parameters of the input models onto the blend model. Once the complete network has been composed, the parameter values are bound and information is continuously propagated while dynamically running the network. We will next discuss the computational theory for blending, then examine the double scope binding configuration applied to audio synthesis, video editing, and presentation mining.

Computational Theory

The blending framework for discovering emergent semantics in media consists of three main components: **ontologies** that provide a shared description of the domain; **operators** that apply transformations to the inputs and perform computations on the input models; and an **integration mechanism** that helps the user discover emergent structure in the media.

Ontologies

Ontologies are a key enabling technology for semantic media. An ontology may be defined as a formal and consensual specification of a conceptualization that provides a shared understanding of a domain. Moreover, the ontology provides an understanding that can be communicated across people and application systems. Ontologies may be of several types ranging from the conceptual specification of a domain to an encoding of computer programs and their relationships. In addition to providing a structured representation, ontologies offer the promise of a shared and common understanding of a domain that can be communicated between people and application systems. Thus, the use of ontologies brings together two essential elements for discovering semantics in media:

- Ontologies define a formal semantics for information that can be processed by a computer.
- Ontologies define real-world semantics that facilitates the linking of machine processable content with user-centric meaning.

Ontologies are used in the media blending framework to model relationships among media elements, as well as provide a domain model for user-centric operators. This provides a level of abstraction that is critical for model-based approaches to media synthesis. As we have seen in the sound synthesis models and automated video editing examples; the input media may come from disparate sources, be described by differing metadata schema, and pose unique processing constraints. Moreover, the intended usage of the media is likely to be different from the initial composition and annotation. Thus the ontologies provide a critical link between the end user and the computer which is necessary for emergent semantics.

Operators

The linkages among the two input spaces and the media blend in Figure 5 are supported by a set of core operators. Two of these operators are called *projection* and

compression. *Projection* is the process in which information in one space is mapped onto corresponding information in another space. In the video editing domain, projection occurs through the mapping of temporal structures from music onto the duration and sequencing of video clips. The hierarchical structure of music and the editing instructions for the video can both be modeled as a graph. Since each model is represented by a graph structure, projection amounts to a form of graph mapping. In general, the mapping between models is not direct, so the ontology from the generic space is used to construct the transformation that maps information between input spaces.

Compression is the process in which detail from the input spaces is removed in the blend space in order to provide a condensed description that can be easily manipulated. Compression is achieved in an audio morph through the low dimensional control parameters for the transformation between to two input sounds. The system performs these operations in the blended space and projects the results back to any of the available input spaces.

Integration Mechanism

The consequence of defining the operators for projection and compression is that a new integration process called, *running the blend* becomes possible within this framework (Fauconnier, 1997). Running the blend is the process of reversing the direction of causality, thereby using the blend to drive the production of inferences in either of the input spaces. In the blend space of the video editing example, the duration of an edited video clip is related to the loudness of the music and the start and stop times are determined by salient beats in the music. The process of running the blend causes these constraints to propagate back to the music input model to determine the loudness value and the timing of salient beats that satisfy the editing logic of the music video blend. Thus the process of running the blend means that operations applied in the blend model are projected back to the inputs to derive emergent semantics, such as music driven video editing.

In the case of mining presentations for information, preprocessing by the system analyzes the textbook to extract terms and relations, which are then added as concept instances of a domain ontology within the textbook model. The user seeks to query the courseware for information that combines the temporal sequencing of the video lecture models with the structured organization of the textbook model. This integrated view of the media is constructed by invoking a blending model, such as '*find path*' or '*find similar*' to translate the user query into primitives that are suitable for the input models. Once the lecture presentation network has been constructed, the user can run the blend to query the input models for a path through the lecture video that links any two given topics from the textbook. The integration mechanism of this blend provides a parameterized model of a path that can be used to navigate through the media, mine for relationships, or compose answers to the original query. The emergent semantics of this media blending model is a path through the video content that exhibits relationships derived from the textbook. Due to the double scope binding of this network, the blending model can also be used to project information from the video onto the textbook, thereby imposing the temporal sequencing of the video presentation onto the hierarchical organization of the textbook. Additional blending networks, such as the 'find similar' blend, can be used to integrate information from the two input sources to discover similarities.

Thus, the integration mechanism for multimedia presentations produces a path in the blend model. The system can then 'run the blend' on the path using all of the emergent properties of the path, such as contracting, expanding, branching, and measuring distance. The specialized domain operators can now be applied and their consequences projected back onto the input spaces. The blend model for presentations encodes strategies that the user may use to locate information, discover the relationships between two topics, or recover from a comprehension failure while viewing the video presentation.

EMERGENT MEDIA SEMANTICS

The ability to derive emergent semantics in the media blending framework depends upon the specification of an ontology, the definition of operators, and processing the integration network to run the blend. This section draws upon the domains of audio synthesis, video editing, and media mining to illustrate the key features of this framework.

The combination of a reference ontology with synthesis models facilitates the manipulation of media semantics. The example of an audio morph between two input audio models illustrates the use of model semantics to provide the cross-mapping between models and the use of the domain ontology to discover emergent structure.

The blending framework for video editing employs a domain specific ontology and defines a set of operators that are applied to the configuration of models illustrated in Figure 5. The key operators, as defined in Conceptual Integration Theory (Fouconnier, 1997), enable the construction of a metaphorical space that contains a mapping of selected properties from each of the input models. The emergent semantics of the edited music video is derived from the cross space mappings of the blending network.

The effective use of emergent semantics is obtained through the processing of the blend model. The final part of this chapter uses media mining in the e-learning domain to provide a detailed example for the integration of a domain ontology, operators, and a query mechanism that runs the blend. This shows that the blending framework provides a systematic way to manage the media semantics that emerges from heterogeneous media.

The Audio Morph

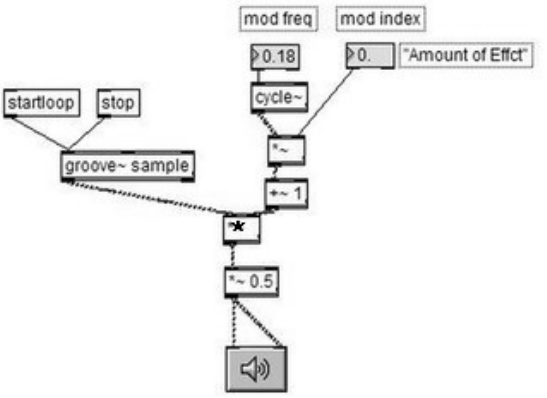
Model based audio synthesis is based on a collection of primitive units and functional modules "patched" together to build sound models. Units include oscillators, noise generators, filters, event generators, and arithmetic operators. Units have certain types of signals and ranges they expect at its inputs, and produce certain types of signal at their respective outputs. Some units can be classed together (e.g. signal generators), and many bear other types of relationships to one another. The composition of units and modules into sound models is informed and constrained by their parameterizations and the relations between entities which can be specified as a domain ontology. The ontology might include information such as the fact that an oscillator takes a frequency argument, that a sine wave oscillator "is-a" oscillator, and that the "null condition" (when a transform unit has no effect, or the output of a generator is constant) for an oscillator occurs when its frequency is set to 0. Experts have implicit knowledge of these ontologies that they use to build and manipulate model structures. There have been attempts to make these ontologies explicit so that non-experts would have support in achieving their semantically specified modeling objectives (Slaney, Covell, & Lassiter, 1996).

Synthesis algorithms are typically represented as a signal flow diagram, as in the MAX/MSP software package. An example of a simple "patch" for sinusoidal amplitude modulation of a stored audio signal is shown in Figure 6.

The representation of datatype, functional, and relational information within a domain ontology of sound synthesis unites provides a new rich source of information that can be queried and mined for hints on how to transform or to build new models. The model components and structure define and constrain the behavior of the sound generating objects and relate the dynamics of parametric control to changes in the sound. They do not themselves determine the semantics, but are more closely allied to semantic descriptions than a "flat" audio stream. A model component ontology would thus be useful for making associations between model structures and the semantics that models are given by designers or by the contexts in which their sounds are used. Relationships, such as distances and morphologies between models, could be discovered and exploited based not only on the sounds they produce, but also on the structure of the models that produce the sounds. Applications of these capabilities include model database management and tools for semantically-driven model building and manipulation.

The availability of a domain ontology along with model building and manipulation tools provides the key resources for discovering emergent semantics among sound models. An example of emergent semantics comes from the media "morph". Most people are familiar with the concept in the realm of graphics, perhaps less so with sound. A "morph" is the process of smoothly changing one object into another over a period of time. There are several issues that make this deceptively simple concept challenging to implement in sound, not the least of which is that the individual source and target sound "objects" may themselves be evolving in time. Also, two objects are not enough to determine a morph — they must both have representations in a common space so that a path of intermediate points can be defined. Finally, given two objects in a consistent

Figure 6. The central multiplication operation puts a sinusoidal amplitude modulation (coming from the right branch) on to the stored "sample" signal on the left branch in this simple MAX/MSP "patch".



representational space, there are typically an infinite number of paths that can be traversed to reach one from the other, some of which will be effective in a given usage context, others possibly not.

The work on this issue has tended to focus on various ways of interpolating between spectral shapes of recorded sounds (Slaney et al., 1996). This approach works well when the source and target sounds are static so that the sounds can be transformed into a spectral representation. Similar to the case with graphical morphs, corresponding points can be identified on the two objects in this space. A combination of space warping and interpolation are used to move between the target and source.

A much deeper and informative representation of a sound is provided in terms of a *sound model*. There are several different ways that models can be used to define a morph. The more the model structures can be exploited, the richer are the possible emergent semantics.

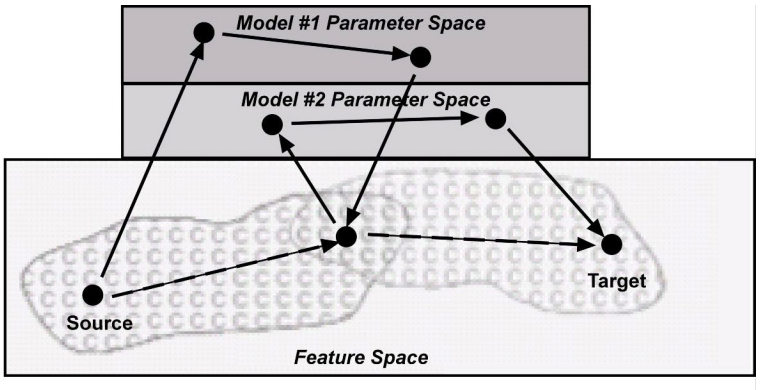
If two different sounds can be generated by the same model, then a morph can be trivially defined by selecting a path from the parameter setting that generates one to a setting that generates the other. In this case, the "blend space" is the same as the model for the two sounds, so although the morph may be more interesting than the spectral variety discussed above, no new semantics can be said to emerge.

If we are given two sounds, each with a separate model capable of generating the sounds, then the challenge is to find a common representational space in which to create a path that connects the two sound objects. One possible solution would be to define a set of "feature detectors" (e.g., spectral measurements, pitch, measures of noisiness) that would provide a kind of description of any sound. This solves the problem of finding a common space in which both source and target can be represented. Next a region of the feature space that the two model sound classes have in common needs to be identified, and paths from the source and target need to be delineated such that they intersect in that region. If the model ranges don't intersect in the feature space, then a series of models with ranges that form a connected subspace needs to be created to support such a path so that a "morph" can be built using a series of models as illustrated in Figure 7. This process requires knowledge about the sound generation capabilities of each model at a given point in feature space.

We mentioned earlier that a model is defined not only by the sounds within its range, but in the paths it can take through the range as determined by the control parameterizations. The dynamic behavior defined by the possible paths play a key role in any semantics the model might be given. The connected feature space region defines a path between the source and target sounds in a particular way that will create and constrain a semantic interpretation. However, in this case, the new "model" is less than satisfying because as a combination of other models, only one of which is active at a time, it can't actually generate sounds that were not possible with the extant models. Moreover, if the kludging together of models is actually perceived as such, then new semantics fail to arise.

Another way to solve the problem would be to embed the two different models in to a blended structure where each original model can be viewed as a special case given by specific parameter settings of the meta-model. This could be done trivially by building a meta-model that merely "mixes" the audio output from each model separately, with a parameter that controls the relative contribution from each submodel. Again we have a

Figure 7. A morph in feature space performed using one model that can generate the source sound, another model that can generate the target sound, and a path passing through a point in feature space that both models are capable of generating. If the source-generating model and the target-generating model do not overlap in feature space, intermediate models can be used so that a connected path through feature space is covered.

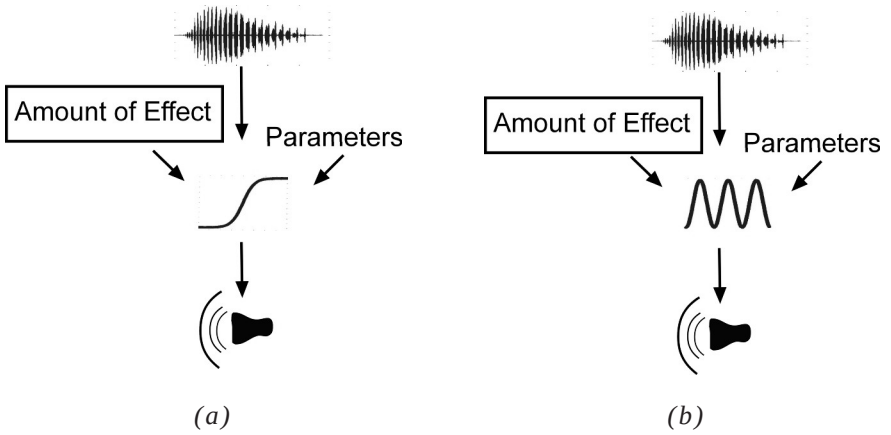


trivial "morph" that would not be very satisfying, and because sound mixes from independent sources are generally perceived as mixes rather than as a unified sound by a single source, the semantics of the individual component models would presumably be clearly perceptible.

There are, however, much richer ways of embedding two models into a blended structure such that each submodel is a sufficient description of the meta-model under specific parameter settings. The blended structure wraps the two submodels that generate the morphing source and target sounds and exposes a single reduced set of parameters. There must exist at least one setting for the meta-model parameters such that the original morphing target sound is produced, and one such that the original morphing source sound is produced in order to create the transformation from source sound to target sound. The meta-model parameterization defines the common space in which both the original sounds exist and in which any number of paths may be constructed connecting the two. We discussed this situation earlier, except in this case, the meta-model is genuinely new, and has its own set of capabilities and constraints defined by the relationship between the structure of the two original models, but present in neither. New semantics emerge from the domain ontology, mappings between models, and the integration network created in the blend.

As a concrete example of an audio morph with emergent semantics, consider two different sounds: one the result of waveshaping on a sinusoid, the other the result of amplitude modulation of a sampled noise source as illustrated in Figure 8. Each structure creates a distinctive kind of distortion of the input signal. One way of combining these two models into a meta-model is shown in Figure 9. To combine these two models, we use knowledge about the constituent components of the models, which could be exploited automatically if they were represented as a formal ontology as discussed above. In

Figure 8. Two kinds of signal distortion. a) This patch puts the recorded sample through a non-linear transfer function (tanh). The "amount of effect" determines how non-linear the shaping is, with zero causing the original sample to be heard unchanged. b) A sinusoidal amplitude modulation of the recorded sample.

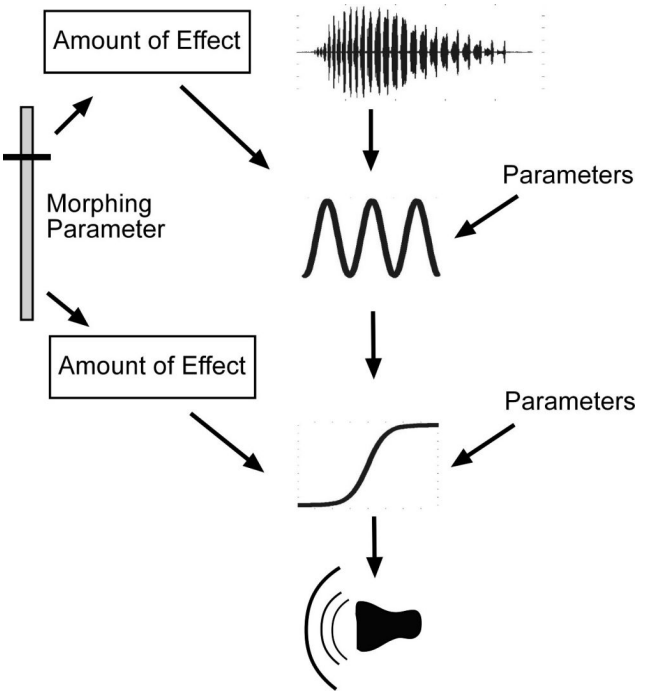


particular, knowing the input and output types and ranges for signal modifying units, and knowing specific parameter values for which the modifiers have no effect on the signal (the "null" condition), we can structure the model for morphing. Knowledge of null conditions, in particular, was used so that the effect of one submodel on the other would be nullified at the extreme values of the morphing parameter. Using knowledge of the modifier unit's signal range expectations and transformations permits the models to be integrated at a much deeper structural level than treating the models as black boxes would permit.

Most importantly, blending the individual model structures creates a genuinely new model capable of a wide range of sounds that neither submodel was capable of generating alone, yet including the specific sounds from each submodel that were the source and the target sounds for the morph. A new range of sounds implies new semantic possibilities.

New semantics can be said to arise in another aspect as well. In the particular blend illustrated above, most of the parameters exposed by the original submodels are still available for independent control. At the extreme values for the morphing parameter, the original controls have the same effect that they had in their original context. However, at the "in between" values for the morphing parameter, the controls from the submodel have an effect on the sound output that is entirely new and dependent upon the particular submodel blend that is constructed. This emergent property is not present in the trivial morph described earlier which merely mixed the audio output of the two submodels individually. Since a morph between two objects is not completely determined by the endpoints, but by the entire path through the blend space, there is a creative role for a "human-in-the-loop" to complete the specification of the morph according the usage context.

Figure 9. Both the waveshaping (WS) and the amplitude modulation (AM) models embedded in a single "meta-model". When the "WS vs. AM" morphing parameter is at one extreme or the other we get only the effect of either the WS or the AM model individually. When the morph parameter is at an in between state, we get a variety of new combinations of waveshaping of the AM signal and/or amplitude modulation of the waveshaped signal (depending on the other parameter settings).



We have shown, that given knowledge about how elementary units function within models in the form of an ontology, structures for different models can be combined in such a way that gives rise to new sound ranges and new handles for control. Semantics emerge that are related to those of the model constituents, but in rich and complex ways. There are, in general, many ways that sound models may be combined to form new structures. Some combinations may work better in certain contexts than others. How desired semantics can be used to guide the construction process is a topic that warrants further study.

The Video Edit

The proliferation of digital video cameras has enabled the casual user to easily create recordings of events. However, these raw recordings are of limited value unless they are edited. Unfortunately, manual editing of home videos requires considerable time and skill to create a compelling result. During many decades of experimentation, the film industry has developed a grammar for the composition of space, time, image, music, and

sound that are routinely used to edit film (Sharff, 1982). The mechanisms that underlie cinematic editing can be described as a blending network that leverages the cognitive perceptions of audio and imagery to create a compelling story. The Video Edit example borrows from such cinematic editing techniques to construct a blending network for the semi-automatic editing of home video. In this network, the generic model is an encoding of the cinematic editing rules relating music and video, and the input models represent structural features of the music and visual events in the video.

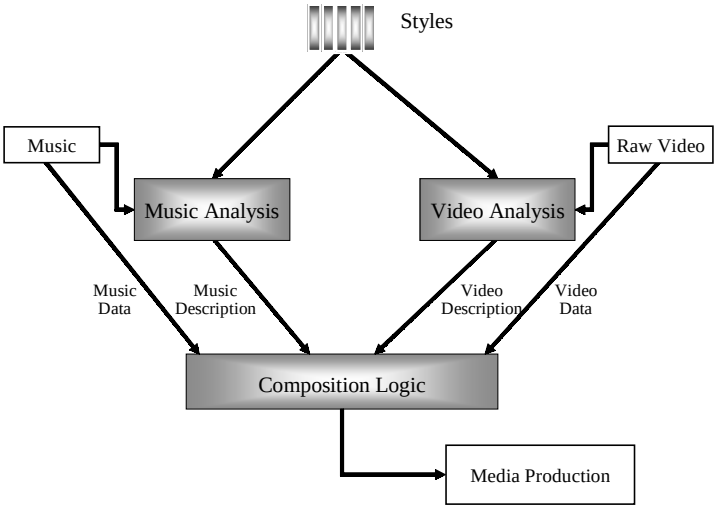
Encoding of aesthetic decisions for editing video to music is key for creating the blending model. Traditional film production techniques start by creating the video track, then add sound effects and music to enhance the affective qualities of the video. This is a highly labor intensive process that does not lend itself well to automation. In the case of the casual user with a raw home video, the preferred editing commands emphasize functional operations, such as selecting the overall style of cinematic editing, choosing to emphasize people in the video, selecting the music, and deciding how much native audio to include in the final production.

The generic model for the music and video inputs is a collection of editing units that describe simple relations between fragments of audio and video. Each unit captures partial information associated with a cinematic editing rule, thus the units in the generic model can be composed in a graph structure to form more complex editing logic. One example of an insertion unit specifies that the length of a video clip to be inserted should be inversely proportional to the loudness of the music. During the construction of the blending network the variables for video length and music loudness are bound and specific values are propagated during the subsequent "running of the blend" to dynamically produce the final edited video. Another example of a transition unit specifies how two video clips are to be spliced together. When this unit is added to the graph structure, it specifies the type of transition between video clips, the duration of the transition, and the inclusion of audio or graphical special effects. Yet another insertion unit may relate the timing and visual characteristics of people in the video to various structural features in the music. The generic model may therefore be viewed as an ontology of simple editing units that can be composed into a graph structure by the blending model for subsequent editing of the music and video inputs.

The video input model contains the raw video footage plus the shots detected in the video, where each shot is a sequence of contiguous video frames containing similar images. The raw video frames are then analyzed in terms of features for color, texture, motion, as well as simple models for the existence of people in the shot or other salient events. This analysis provides the metadata for a model representing the input video for use in the subsequent video editing. For example, the video model includes techniques for finding parts of the shots which contain human faces. The information about faces can be combined with the editing logic to create a final production which emphasizes the people in the video. In this way, the system can automatically construct a people-oriented model of the input video.

The model for the input music needs to support the editing logic of the generic model for cinematic editing and the high level commands from the user interface. The basic music model is composed of a number of parameters, including the tempo, rhythm, and loudness envelope for the music. The combined inputs of the video model and the music model in Figure 10 are integrated with the cinematic styles in the blending model to

Figure 10. The blending network for automatic editing of video according to the affective structures in the music and operators for different cinematic styles.



produce a series of editing decisions — when to make cuts, which kinds of transitions to use, what effects to add, and when to add them.

The composition logic in the blend model integrates information from three places: a video description produced by the video analysis, a music description produced by the music analysis, and information about the desired editing style as selected by the user. The composition logic uses the blending model to combine these three inputs in order to make the best possible production from the given material — one which is as stylish and artistically pleasing as possible. It does this by representing the blended media construction as a graph structure and opportunistically selecting content to complete the media graph. This process results in the emergent semantics of a "music video" which inherits partial semantics from the music and from the video.

The Presentation

The Presentation example illustrates the use of emergent structure to facilitate information retrieval from online e-learning courseware. A simple form of emergent structure is a path that combines concept relationships from a textbook with temporal sequencing from the video presentation. The path structure can then be manipulated to gain insight into the informational content of the courseware by performing all of the standard operations afforded by paths, such as traversal, compression, expansion, branching, and the measurement of distance.

Find Path Scenario

As distance education expands by placing ever larger quantities of course content online, students and instructors have increasing difficulty navigating the courseware and assimilating complex relationships from the multimedia resources. The separation in space and time between students and teachers also makes it difficult to effectively formulate questions that would resolve media based comprehension failures. The 'find path' scenario addresses this problem by combining media blending with information retrieval to assist the student in formulating complex queries about lecture video presentations.

Suppose that a student is half way through a computer science course on the "Introduction to Algorithms", which teaches the computational theory for popular search optimization techniques (Corman, Leiserson, Rivest, & Stein, 2001). The topic of Dynamic Programming was introduced early in the course, then after several intervening topics the current topic of Greedy Algorithms is presented. The student realizes that these two temporally distant topics are connected, but the relationship is not evident from the lecture presentations due to the temporal separation and the web of intervening dependencies between topics. To resolve this comprehension failure, the student would like to compose a simple query, such as "*Find a path linking greedy algorithms to dynamic programming*" and have the presentation system identify a sequence of locations in the lecture video which can be composed to provide an answer to the query.

Note that the path that the student is seeking does not exist in either the textbook or the lecture video. The textbook contains a hierarchical organization of topics and instances of concept relations. The sequencing of topics from the beginning of the book to the end provides a linear ordering of topics, but not necessarily a chronological ordering. The lecture videos do not contain the semantics of a path since they have a purely chronological sequencing of content. As we shall see, partial structure from the textbook and the video must be projected onto the blend to create the emergent structure of the path.

Media Blending Network

Traditional approaches to media integration are based on the use of a generic model to provide a unified indexing scheme for the input media plus cross media mapping (Benitez & Chang, 2002). The media blending approach adds the blend model as an integral part of the integration mechanism. This has two consequences. Firstly, the blend model adds considerable richness to the media semantics in terms of the operators that can be applied, such as projection, compression, and the propagation of information back into the input spaces. Secondly, by explicitly providing for emergent structures in the computational framework, we can potentially achieve a higher level of integration of multimedia resources. Of particular interest for the Presentation network is the semantics of a path that emerges from the media blend. Once the path is obtained, then all of the common operations on paths, such as expand, compress, extend, append, branch, and the measurement of distance can be applied to the selected media elements.

The configuration of models used to construct the Presentation network is illustrated in Figure 11. The *textbook model* provides one input to the network. This model contains instances of terms and relations between terms that can be extracted through standard text processing techniques. The instances of terms and relations are mapped

onto the abstract concepts of the domain ontology. The ontology is subsequently converted to a graph structure for efficient search. The textbook has explicit structure due to the hierarchical organization of chapters and topics, as well as the table of contents, and index. There is also implicit structure in the linear sequencing of topics and the convention among textbooks that simpler material comes before more complex material.

The *lecture model* provides the second input which represents the lecture video, transcripts, and accompanying slide presentation. The metadata for the lecture model can be derived automatically through the analysis of perceptual events in the video to classify shots according to the activity of the instructor. The text of the transcripts can be analyzed to extract terms and indexed for text based queries. The slide presentation and associated video time stamps provide an additional source of key terms and images that can be cross mapped to the textbook model.

The *generic model* for the Presentation network contains the core domain ontology of terms and relations used in the course. As we shall see later, the cross-space mapping between the textbook model and the lecture model occurs at the level of extracted terms and their locations in the respective media. The concepts in the core ontology thus provide a unified indexing scheme for the term instances that occur as a result of media processing in the two input models.

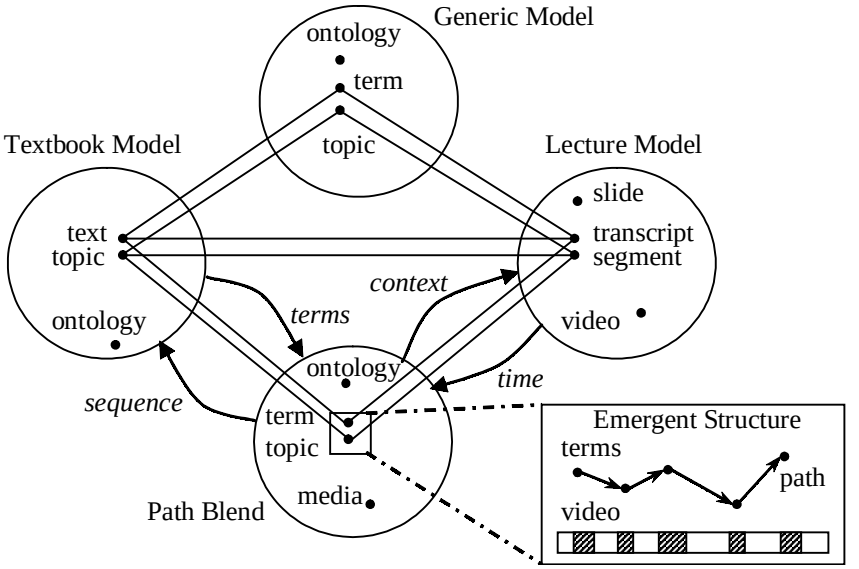
The *blend model* for the 'find path' scenario receives projections of temporal sequence information from the lecture video and term relations from the textbook. When the user issues a query to find a path in the lecture video that goes from topic A to topic B, the blend model first accesses the textbook model to expand the query terms associated with the topics A and B. The graph representation of the textbook model is then searched for a path linking these topics. Once a path is found among the textbook terms, the original query is expanded into a sequence of lecture video queries, one for each of the terms plus local context from the textbook. The blend model then evaluates each video query and assembles the selected video clips into the final path structure (see insert in Figure 11). At this point, the blend model has fully instantiated the path as a blend of the two inputs.

Once the path has been constructed, the user can run the blend to perform various operations on the path. Note that the blending network has added mappings between the input models, but has not modified the original models. Thus, the path blend can, for instance, be used to project the temporal sequencing from the time stamps of the lecture video back onto the textbook model to construct a navigational path in the textbook with sequential dependencies from the video.

Operators

Mappings between models in the Presentation network in Figure 11 support a set of core operators for information retrieval in mixed media. Two of these operators are called *projection* and *compression*. As seen in previous examples, projection is the process in which information in one model is mapped onto corresponding information in another model. Since both input models are represented by a graph structure, where links between nodes are relations, projection between inputs amounts to a form of graph matching to identify corresponding elements. These elements are then bound so that information can pass directly between the models. A second source of binding occurs between each input model and the emergent structure that is constructed in the blend

Figure 11. Media integration in the Presentation network for the 'Find Path' blend.



model. This double scope binding enables the efficient projection of information within the network.

Compression is another core operator of the blending network that supports media management through semantics. For example, traditional methods for constructing a video summary require the application of specialized filters to identify relevant video segments. The segments are then composed to form the final summary. Instead of operating directly on the input media, compression operates on the emergent path structure and projects the results back to the input media. Thus by operating on the path blend, one can derive the shortest time path, the most densely connected path, or the path with the fewest definitions from the lecture video. The system performs these operations on the blended model and projects the results back to either of the available input models to determine the query result.

The consequence of defining the operators for projection and compression is that a new process called, *running the blend* becomes possible within this framework. Running the blend is the process of reversing the direction of causality within the network, thereby using the blend to drive the production of inferences in either of the input spaces. In the 'find path' example, the application of projection and compression on the path blend means that the user can manage the media using higher level semantics. Moreover, all of the standard operations on paths, such as contracting, expanding, reversing, etc. can now be performed and their consequences projected back onto the input spaces. Finally, the user can perform a series of queries in the blend and project the results back to the inputs to view the results.

Integration Mechanism

The network of models in the Presentation blend provides an integration mechanism for the multimedia resources. Once the network is constructed, it is possible to process user queries by running the blend. In the find path scenario, the student began with a request to find a relationship between Dynamic Programming (DP) and Greedy Algorithms (GA). The system searches the domain ontology of the textbook model to discover a set of possible paths linking DP with GA, subject to user preferences and event descriptions. The user preferences, event descriptions, and relations among the path nodes in the ontology are used to formulate a focused search for similar content in the video presentation. The resultant temporal sequence of video segments is added to the emergent path structure in the blend model.

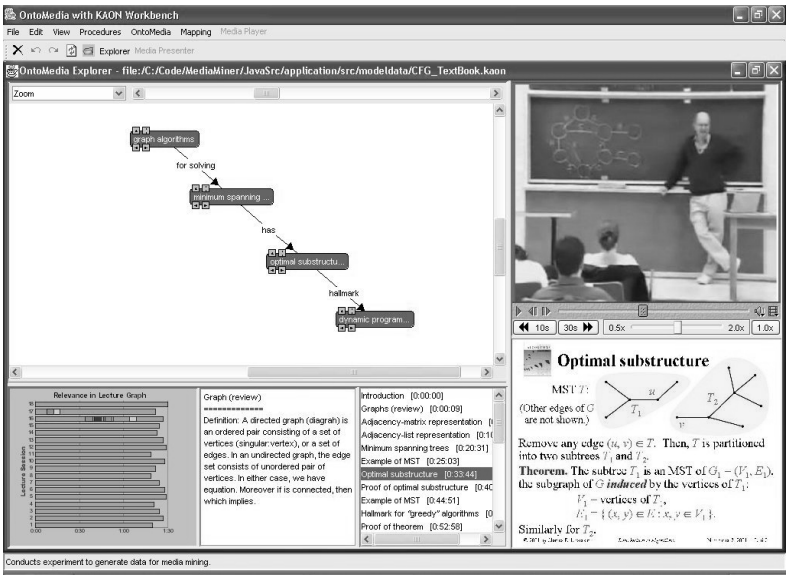
As discussed previously, when the student requested to find a path from topic DP to topic GA, a conceptual blend was formed which combined the ontology from the textbook with the temporal sequencing of topics from the lecture. The result was a chronological path through the sequence of topics linking DP to GA. This path can now be used in an intuitive way to compress time, expand the detail, select alternative routes, or combine with another path. The resultant path through a sequence of interrelated media segments in the '*find path*' blend is the emergent structure arising from the processing of the user's query. Thus, one can now start from the constructed path and project information back onto the input spaces to mine for additional information that was previously inaccessible. For example, one could use the path to select a sequence of text locations in the textbook that correspond to the same chronological presentation of the topics that occurs in the lecture. Thus, the blending network effectively uses the instructor's knowledge about the pedagogical sequencing of topics to provide a navigational guide through the textbook.

We have designed a system for indexing mixed media content using text, audio, video, and slides and the segmentation of the content into various lecture components. The GUI for the ontology based exploration and navigation of lecture videos is shown in Figure 12. By manipulating these lecture components in the media blend, we are able to present media information to support insight generation and aid in the recovery of comprehension failures during the viewing of lecture videos. This dynamic composition of cross media blends provides a malleable representation for generating insights into the media content by allowing the user to manage the media through the high level semantics of the media blend.

FUTURE TRENDS

The development of the Internet and the World Wide Web has lead to the globalization of text based exchanges of information. The subsequent use of web services for the automatic generation of web pages from databases for both human and machine communication is being facilitated by the development of Semantic Web technologies. Similarly, we now have the capacity to capture and share multimedia content on a large scale. Clearly, the plethora of pre-existing digital media and the popularization of multimedia applications among non-professional users will drive the demand for authoring tools that provide a high level of automation.

Figure 12. User interface for the Presentation network. Display contains the following frames (clockwise from top left corner): Path Finder, Video Player, Slide, Slide Index, Textbook display, and a multiple timeline display of search results presented as gradient color hotspots on a bar chart.



Preliminary attempts toward the use of models for the generation of sound effects for games and film, as well as the retrieval of video from databases has been primarily directed toward human-to-human communication. The increasing use of generative models for media synthesis and the ability to dynamically construct networks for combining these models will create new ways for people to experience media. Since the semantics of the media is not fixed, but arises from the media and the way that it is used, the discovery of emergent semantics through ontology based operations is becoming a significant trend in multimedia research. The convergence of generative models, automation, and ontologies will also facilitate the exchange of media information between machines and support the development of a media semantic web.

In order to realize these goals further progress is needed in the following technologies:

- Tools to support the development of generative models for media synthesis.
- Use of semantic descriptions for the composition of models into blending networks.
- Automation of user tasks based on the mediation between blending networks as an extension to the ongoing research in database schema mediation.
- Formalization of ontologies for domain knowledge, synthesis units, and relationships among generative models.

- Discovery and generation of emergent semantics.

The trend toward increasing the automation of media production through the creation of media models relies upon the ability to manage the semantics that emerges from user-centric operations on the media.

CONCLUSIONS

In this chapter we have presented a framework for media blending that has proved useful for discovering emergent semantics. Concrete examples drawn from the domains of video editing, sound synthesis and the exploration of multimedia content for lecture based courseware have been used to illustrate the key components of the framework. Ontologies for sound synthesis components and the perceptual relations among sounds were used to describe how emergent properties arise from the morphing of two audio models into a new model. From the domain of automatic home video editing, we have described how the basic operators of projection and compression lead to the emergence of a stylistically edited music video with combined semantics of the source music and video. In the video presentation example, we have shown how multiple media specific ontologies can be used to transform high level user queries into detailed searches in the target media.

In each of the above cases, the discovery and/or generation of emergent semantics involved the integration of descriptions from four distinct spaces. The two input spaces contain the models and metadata descriptions of the source media that are to be combined. The generic space contains the domain specific information and mappings that relate elements in the two input spaces. Finally, the blend space is where the real work occurs for combining information from the other spaces to generate a new production according to audio synthesis designs, cinematic editing rules, or navigational paths in presentations as discussed in this chapter.

REFERENCES

Benitez, A. B., & Chang, S. F. (2002). Multimedia knowledge integration, summarization and evaluation. *Proceedings of the 2002 International Workshop on Multimedia Data Mining*, Edmonton, Alberta, Canada, (pp. 39-50).

Corman, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to Algorithms*. Cambridge, MA: MIT Press.

Davis, M. (1995). Media Streams: An iconic visual language for video Representation. *Readings in Human-Computer Interaction: Toward the Year 2000.* Baecher, R. M., Grudin, J., Buxton, W. A. S., & Greenberg, S. (Eds.), 2nd ed., 854-866. San Francisco: Morgan Kaufmann Publishers Inc.

Dorai, C., Kermani, P., & Stewart, A. (2001, October). E-Learning media navigator. *Proceedings of the 9th ACM International Conference on Multimedia*. Ottawa, Canada, (pp. 634-635).

Fauconnier, G. (1997). *Mappings in thought and language*. Cambridge, UK: Cambridge University Press.

Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., et al. (2004, August). Modeling by example. *ACM Transactions on Graphics (SIGGRAPH 2004)*.

Kovar, L., & Gleicher, M. (2003). Flexible automatic motion blending with registration curves. *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, San Diego, California, (pp. 214-224).

Kellock, P., & Altman, E. J. (2000). *System and method for media production*. Patent WO 02/052565.

Mott, R. L. (1990). *Sound Effects: Radio, TV and film*. Focal Press.

Nack, F., & Hardman, L. (2002). *Towards a syntax for multimedia semantics.* CWI Technical Report, INS-R0204, April.

Rolland, P.-Y., & Pachet, F. (1995). Modeling and applying the knowledge of synthesizer patch programmers. In G. Widmer (Ed.), *Proceedings of the IJCAI-95 International Workshop on Artificial Intelligence and Music, 14th International Joint Conference on Artificial Intelligence*, Montreal, Canada. Retrieved June 1, 2004: *http://citeseer.ist.psu.edu/article/rolland95modeling.html*

Russolo, L. (1916). *The art of noises*. Barclay Brown (translation). New York: Pendragon Press.

Santini, S., Gupta, A., & Jain, R. (2001). Emergent semantics through interaction in image databases. *IEEE Transaction of Knowledge and Data Engineering*, 337-351.

Sharff, S. (1982). *The elements of cinema: Toward a theory of cinesthetic impact*. New York: Columbia University Press.

Slaney, M., Covell, M., & Lassiter, B. (1996). *Automatic audio morphing.* Proceedings of IEEE International Conference Acoustics, Speech and Signal Processing, Atlanta, 1-4. Retrieved June 1, 2004: *http://citeseer.nj.nec.com/slaney95automatic.html*

Smalley, D. (1997). Spectromorphology: Explaining sound shapes. *Organized Sound, 2*(2), 107-126.

Staab, S., Maedche, A., Nack, F., Santini, S., & Steels, L. (2002). Emergent semantics. *IEEE Intelligent Systems: Trends & Controversies, 17*(1), 78-86.

Thompson Learning (n.d.). Retrieved June 1, 2004: *http://www.thompson.com/*

Veale, T., & O'Donoghue, T. (2000). Computation and blending. *Cognitive Linguistics, 11*, 253-281.

WebCT (n.d.). Retrieved June 1, 2004: *http://www.webct.com*